



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/696,964	10/30/2003	Neelam N. Vaidya	5681-54200	8464
58467 MHKKG/SUN P.O. BOX 398 AUSTIN, TX 78767	7590 08/06/2009			
EXAMINER				
WU, JUNCHUN				
ART UNIT		PAPER NUMBER		
2191				
NOTIFICATION DATE		DELIVERY MODE		
08/06/2009		ELECTRONIC		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patent\_docketing@intprop.com  
ptomhkk@gmail.com

### Office Action Summary

**Application No.**

10/696,964

**Applicant(s)**

VAIDYA, NEELAM N.

**Examiner**

JUNCHUN WU

**Art Unit**

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 02 May 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-22 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-22 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SF/ICE)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

**DETAILED ACTION**

1. This office action is in response to amendment filed on May 2, 2009.
2. Claims 1-22 are pending.

***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-2, 5, 11-13, 16, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chamberlain et al. (US Patent No. 6,434,744 B1, hereinafter "Chamberlain"), and in view of Curtis (US Patent No. 6,601,236 B1, hereafter "Curtis").

5. Per claims 1 and 12

Chamberlain discloses

- a method or system for patching applications, comprising: deploying a patch package on a first computer running a first type of operating system, wherein the patch package comprises a patching mechanism and a first set of one or more new code components (col.5 lines 10-12 "*for patching a software application and providing an installer program with notice of a patch's existence...*" & lines 19-23 "*the present invention provides a mechanism by which the launching of a patch file, termed a "patch package," is passed to the installer program and the installer program takes control of applying the*

*patch.”). Chamberlain further discloses (col.8 lines 52-67 “FIG. 5 provides a functional overview of the several information streams within a typical patch package 301...” & col. 11 lines 1-13 “At step 607, the installer program 201 makes the changes to the installation file 233 defined by the transforms 504. For example, the patch may add program files to the application. To that end, the transforms portion 504 may define additional instructions for an “install files” portion 703 of the modified installation file 701 depicted in FIG. 7....”).*

- *executing the patching mechanism on the first computer, wherein executing the patching mechanism comprises replacing a first set of one or more old code components in a first application with the first set of one or more new code components (col.14 lines 30-36 “At step 834, the installer program 201 alters the program file by replacing the affected bits of the program file with the patch bits. This may be accomplished by the installer program 201 executing a patching routine that replaces the affected bits with the new patch bits. Since techniques for applying patch bits to a program file to achieve an altered or updated program file are known to those skilled in the art”).*

But Chamberlain does not explicitly disclose

- *patching mechanism is also executable on a second computer running a second type of operating system.*

However, Curtis discloses

- *The installation program installs files using the cross-platform code, written in a cross-platform language such as Java (col.13 lines 29-42 “The preferred installation program installs files using the cross-platform code, preferably written in a cross-platform*

*language such as Java, and bypasses the need to use the native installation routines, which requires the development of specific installation packages for each operating system on which product will be installed....Further, the preferred embodiment install program 17, which operates in the same manner across different operating system platforms, has the same affect on the operating system as the native installation program.”)*

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teaching of installer program applying the patch (i.e. equivalent to patch mechanism) by Chamberlain with the teachings of Curtis to include the installation program installs files using the cross-platform code, written in a cross-platform language for each operating system on which product will be installed (which is equivalent to “the patching mechanism is also executable on a second computer running a second type of operating system”) in order to provide a method using a cross-platform installation programs that is capable of installing and uninstalling a product on different operating systems and, at the same time, is capable of checking the storage space of the intended destination data storage drives of the operating system (col.3 lines 50-55).

6. Per claims 2 and 13

the rejection of claim 1 is incorporated and Curtis further discloses

- deploying the patch package on the second computer, wherein the patch package further comprises a second set of one or more new code components; and executing the patching mechanism on the second computer, wherein executing the patching mechanism

comprises replacing a second set one or more old code components in a second application with the second set of one or more new code components, wherein the second application is functionally equivalent to the first application (col.8 lines 33-50 “*Preferred embodiments of the invention include a) an installer tool kit, including a system, method, and program, that enables the creation of install programs for any one of a plurality of operating systems as described herein, b) an install program that can be used for multiple platforms to install an application program on a specific operating system; i.e., the same install program can be used to install a similar application program on different operating systems, c) a method for carrying out the functions of the install program, and d) a computer system running the install program on a specific operating system for installing an application program.*” From foregoing discussion, the updating patches may apply to other workstations running on different operating system. )

7. Per claims 5 and 16

the rejection of claim 1 is incorporated and Chamberlain further discloses

- storing the first set of one or more old code components in a separate location before being replaced with the first set of one or more new code components (col.1 lines 17-21 “*The one installer program that comes close to addressing all of a computer user's needs manages the installation of an application so that information related to of each of the application's resources is stored in a "configuration database."*”).

8. For claims 11 and 22

the rejection of claim 1 is incorporated and Curtis further disclose

- The user interface for the patching mechanism is the same on different platforms (col.7 lines 7-19 “*There are certain platform specific operations that are taking place throughout the different objects, i.e., the helper classes 301, install objects 330, etc. Therefore, there is a platform specific module/interface. It is referred to herein as CPP 201. It defines a set of methods, that are representative of different operating systems, for functions that need to take place for an install...*”).

9. Claims 3, 4, 6, 7, 14, 15, 17, and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chamberlain, in view of Curtis and further view of Moshir et al. (US Pub. No. 20020100036 A1, hereafter “Moshir”).

10. Per claims 3, 4, 14 and 15

the rejection of claim 1 is incorporated

But Chamberlain and Curtis do not disclose

- the patching mechanism is a script written in a cross-platform scripting language.

However, Moshir discloses

- the patching mechanism is a script written in a cross-platform scripting language ([0050] “*Programming languages and tools such as Java, Pascal, C++, C, Perl...,*”).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Chamberlain and Curtis and further include the patching mechanism is a script written in a cross-platform scripting language

by the teachings of Moshir in order to use suitable programming language to implement patch by those of skill in the art ([0050]).

11. For claims 6 and 17

the rejection of claim 1 is incorporated

But Chamberlain and Curtis do not disclose

- restoring the first set of one or more old code components to the first application.

However, Moshir discloses

- restoring the first set of one or more old code components to the first application [0065]

*"in some instances when failure is detected the software update is disabled or removed 324 from the target computer, and that machine is returned substantially to its pre-update state or another acceptable (working) non-update state. This may mean that the installed software is taken off the target machine 322; or that not only is the software removed, but all the ancillary files (.dll's, .exe's, etc.) are restored to their pre-update state.").*

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Chamberlain and Curtis and further include restoring the first set of one or more old code components to the first application by the teachings of Moshir in order to provide a mean that target computer was backed up before the software update was installed. When software update is disabled, the backed itself is restored onto machine ([0065]).



12. For claims 7 and 18

the rejection of claim 1 is incorporated

But Chamberlain and Curtis do not disclose

- wherein the patch package further comprises a patch information file, wherein the patch information file comprises information on the first set of one or more new code components and information regarding which application the patch package is applicable to patch.

However, Moshir discloses

- wherein the patch package further comprises a patch information file, wherein the patch information file comprises information on the first set of one or more new code components and information regarding which application the patch package is applicable to patch ([0079] “*The illustrated patch fingerprint comprises one or more general inventory install dependencies 912 that can be used to take a high-level look to see if a specific patch can be installed on a machine. It also includes a signature block 910 that can be used to request specific information from, a target computer 500, and an existence test 908 which can use the signature block information to determine if a specific patch has been loaded on a machine.*”).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Chamberlain and Curtis and further include the patch package further comprises a patch information file, wherein the patch information file comprises information on the first set of one or more new code components and information regarding which application the patch package is applicable

to patch by the teachings of Moshir in order to check the information about the target computer that included software should be present (such as specific version of a program, a patch, a data file or a driver ([0090])).

13. Claims 8-10 & 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chamberlain, in view of Curtis, Moshir and further view of Taylor (US Patent No. 6,161,218).

14. For claims 8 and 19

the rejection of claim 7 is incorporated

Chamberlain discloses

- the patch information file further comprises information on what bugs the patch package is operable to fix (col.1 lines 44-48 “*patching*” refers to the art of modifying or updating an application from one state to another state. Often, patching is performed if an application is in need of a service release or update to remedy a programming bug or other infirmity.”).

But Chamberlain, Curtis and Moshir fail to disclose

- information regarding which other patch packages the patch package is incompatible with.

However, Taylor discloses

- information regarding which other patch packages the patch package is incompatible with [col.7 lines 56-57].
- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine teachings of Chamberlain, Curtis and Moshir and further

include information regarding which other patch packages the patch package is incompatible with by teaching of Taylor in order to scan all the patches in target package's information list and to verify that it is ok to install the patch [Taylor col.7 lines 20-22 & lines 39-41].

15. For claims 9 and 20

the rejection of claim 8 is incorporated

But both Chamberlain and Curtis do not disclose

- storing information on one or more other patch packages that have been previously applied to the first application in a package information file.

However, Moshir discloses

- storing information on one or more other patch packages that have been previously applied to the first application in a package information file ([0128] "*The offsite update server can be configured to store in permanent memory the packages that have already been stored on each target computer. When a new package becomes available, or during the installation of an existing package, existing evidence of the software packages that need to be installed, as well as information about previous installations, is available in some embodiments at the offsite update server 528, and in other instances at the repository site 600.*").
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Chamberlain and Curtis with the teachings of Moshir to include storing information on one or more other patch packages

that have been previously applied to the first application in a package information file in order to check the availability of software patches by receiving a notice from administrator ([0128]).

16. For claims 10 and 21

the rejection of claim 9 is incorporated

But both Chamberlain and Curtis do not disclose

- detecting patch conflicts by comparing the patch information file to the package information file.

However, Moshir discloses

- detecting patch conflicts by comparing the patch information file to the package information file ([0080]) *“These dependencies 912 are compared 808 with information about the target computer 806 previously stored in the inventory library 918. If the install information and the inventory information don't match, then the patch is not installed. In some versions of the invention a message is sent to at least one administrator containing a list of components required (such as necessary hardware and software) for the install.”*

### ***Response to Arguments***

Applicant's arguments filed on May 2, 2009 have been fully considered but they are not persuasive.

**In the remarks, Applicant argues that:**

(a) The cited references fail to teach or suggest (i) patch package comprises a patching mechanism (ii) patching mechanism comprises the patching mechanism performing a replacement of a first set of one or more old code components in a first application with the first set of one or more new code component.

(b) Even were the teachings of the cited art combined, the resulting combination would not teach the specific limitations of current claim.

**Examiner's response:**

(a) Examiner disagrees. First, Chamberlain reference discloses in col.5 lines 9-26 *"The present invention is directed to a system, method, and computer-readable medium for patching a software application and providing an installer program with notice of a patch's existence so that the installer program can modify a configuration database to reflect the patch ... Briefly described, the present invention provides a mechanism by which the launching of a patch file, termed a "patch package," is passed to the installer program and the installer program takes control of applying the patch."* Note that the patch package plus installer program which are stated in the reference are interpreted to be the patch package cited in claims. Applicant appears to be confused with the patch package in the reference with the patch package in the claims.

Second, examiner cited Chamberlain col.14 lines 30-36 *"At step 834, the installer program 201 alters the program file by replacing the affected bits of the program file with the patch bits. This may be accomplished by the installer program 201 executing a patching routine that replaces the affected bits with the new patch bits. Since techniques for applying patch bits to a program file to achieve an altered or updated program file are known to those skilled in the art"* for patching mechanism comprises the patching mechanism performing a replacement of a

first set of one or more old code components in a first application with the first set of one or more new code component as currently amended claim. Examiner elaborate this paragraph by disclosed col.11 lines 46- 67 "The apply patch bits portion 707 is a new portion of the modified installation file 701, i.e., the apply patch bits portion 707 does not appear in the original installation file 233. The apply patch bits portion 707 of the modified installation file 701 defines how the installer program 201 is to alter existing program files with patch bits contained within the cabinets portion 506 of the patch package 301. The apply patch bits portion 707 defines which existing program files require patch bits, and points the installer program 201 to the applicable patch bits within the cabinets portion 506 of the patch package 301. The installer program 201 may contain a routine 340 for performing the actual alteration of the existing program file, or, alternatively, a separate application program (not shown) may be invoked to perform the alteration."

(b) Examiner disagrees.

Chamberlain discloses patch package which includes transforms portion that is how to deploy a patch (col.8 lines 52-67 "*FIG. 5 provides a functional overview of the several information streams within a typical patch package 301 ...*"). The transform information from the patch file provides the installer program with changes (col.3 lines 13-22) & col. 11 lines 1-13 "*At step 607, the installer program 201 makes the changes to the installation file 233 defined by the transforms 504. For example, the patch may add program files to the application. To that end, the transforms portion 504 may define additional instructions for an "install files" portion 703 of the modified installation file 701 depicted in FIG. 7....*"). Also, Chamberlain disclosed (i) patch package comprises a patching mechanism (ii) patching mechanism comprises the patching

mechanism performing a replacement of a first set of one or more old code components in a first application with the first set of one or more new code component. Further, Curtis discloses “The installation program installs files using the cross-platform code, written in a cross-platform language” (which is equivalent to patching mechanism is also executable on a second computer running a second type of operating system) (col.13 lines 29-42) Thus, the combinations of Chamberlain and Curtis teach the limitations of claims 1 and 12.

### ***Conclusion***

17. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Junchun Wu whose telephone number is 571-270-1250. The examiner can normally be reached on 8:00-17:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JW  
/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191